

Common Object Request Broker Architecture C O R B A

Cyber Object has built the Intelligent Customer Advocate System - ICAS utilizing the standards defined by CORBA. Today's software development projects are targeted to heterogeneous computing environments that must integrate new and legacy components developed under different operating systems and programming languages.

Some of the benefits of CORBA include:

- **Supports Development of Multi-tiered Systems** - CORBA's distributed object model enables development of object servers which shelter client applications from direct knowledge of underlying data storage mechanisms such as relational databases.
- **Supports Development in Heterogeneous Environments** - Today's software applications have complex requirements that often require use of many types of computers and tools: GUI builders, desktop computers, servers, etc. CORBA provides a foundation for using these tools and systems together! CORBA simplifies application development by supplying a common programming environment across operating systems.
- **CORBA is Object-Oriented** - CORBA is enabling a new generation of distributed applications to be developed that are based on standards and support remote method invocation on objects.
- **Interoperability** - The CORBA interoperability protocol allows clients running in one ORB to transparently invoke methods on objects within a second ORB. A major benefit of this technology is that applications written to different ORBs can work together, preserving current software investments.
- **Separation of interface and implementation** - CORBA supports this and other principles of object-oriented software development. Interfaces are written in a programming language neutral interface definition language (IDL). This IDL interface is translated to a programming language in the form of client side 'stubs' and server side implementations. Furthermore, in C++ this means that both the stub and implementation classes can share a common base class, allowing application code to be less 'aware' of the distributed nature of the system architecture.

The CORBA Standard

CORBA is a system of standards describing facilities for client-server communication and interaction between distributed software objects. The CORBA standards have been devised by OMG, a large consortium of software and hardware vendors. A number of vendors are producing systems compliant with the CORBA standards for communication between objects, so it seems likely that

CORBA will become an industry standard for client-server systems and distributed software.

The CORBA ORB

The Object Request Broker (ORB) specification is the part of CORBA that describes a "software bus": a mechanism that handles communication between distributed objects. The ORB allows for client-server interaction between heterogeneous objects distributed over a wide-area network, and makes meta information describing the objects in a system and their interfaces available to any object in the system, so that it may access other objects as a client without prior knowledge of their existence. Any object connected to the ORB can play the role of both a client and server object. That is, it can initiate calls to other objects and respond to requests for services from other objects on the ORB.

The ORB specification is programming language, operating system and platform independent. It allows vendors considerable flexibility in their choice of implementation methods. CORBA compliant ORBs are currently available from a number of different vendors based on mechanisms such as RPC, TCP/IP and sockets. Furthermore, the ORB permits transparent communication between objects implemented using a variety of programming languages and operating systems.

The CORBA specification describes protocols for communication between ORBs, which should allow for ORBs provided by different vendors to communicate in a federation. Several vendors are now offering CORBA 2.0 compliant ORBs, though it is not clear that compatibility issues between ORBs from different vendors have yet been resolved.

The Interface Definition Language

The Interface Definition Language (IDL) defined by OMG is a language for describing the interfaces of software objects. According to the CORBA 2.0 specification "an interface is a description of the set of possible operations a client may request of an object" [10]. An interface does not specify the internal data-representation or executable code used to implement an object. In practice, an IDL interface specification may also contain declarations of types, exceptions and constants. In order to facilitate re-use and extensibility of classes, IDL supports multiple inheritances among interface definitions. IDL is independent of programming languages, and may be used to describe objects implemented using a variety of programming languages, compilers or operating systems.

The IDL specification for an object is used to automatically generate "stub" and "skeleton" programs for the object. The stub provides an interface for other client

objects to request services from the object via an ORB. The stub performs tasks such as converting parameters and returned values into a form that allows them to be transmitted via the ORB. The skeleton acts as an interface between an object in its server role and the ORB. IDL specifications can also be used to generate header files in a variety of programming languages, including C++ and Smalltalk, on which implementations can then be based.

In addition, the information represented by the IDL specification for any objects connected to an ORB is compiled and stored in the Interface Repository service that the ORB must provide. The interface repository can be examined by objects on the ORB in order to ascertain what other objects are connected to the ORB and what interfaces they provide. This allows an object to request services from other objects on the ORB without having prior knowledge of the other objects or their interfaces.

CORBA Services

OMG has adopted, or is in the process of adopting, specifications for a number of "services" as part of the CORBA standard. These services consist of descriptions and IDL specifications for modules to perform various common software engineering tasks such as transaction management, concurrency, security and so on. It is hoped that software vendors will provide modules implementing these descriptions, and that software developers will use them, thus allowing users to mix and match software modules and reducing software development time by avoiding duplication. Currently few of the CORBA services have commercial implementations available, and it will be some time before implementations of many of the services become available.